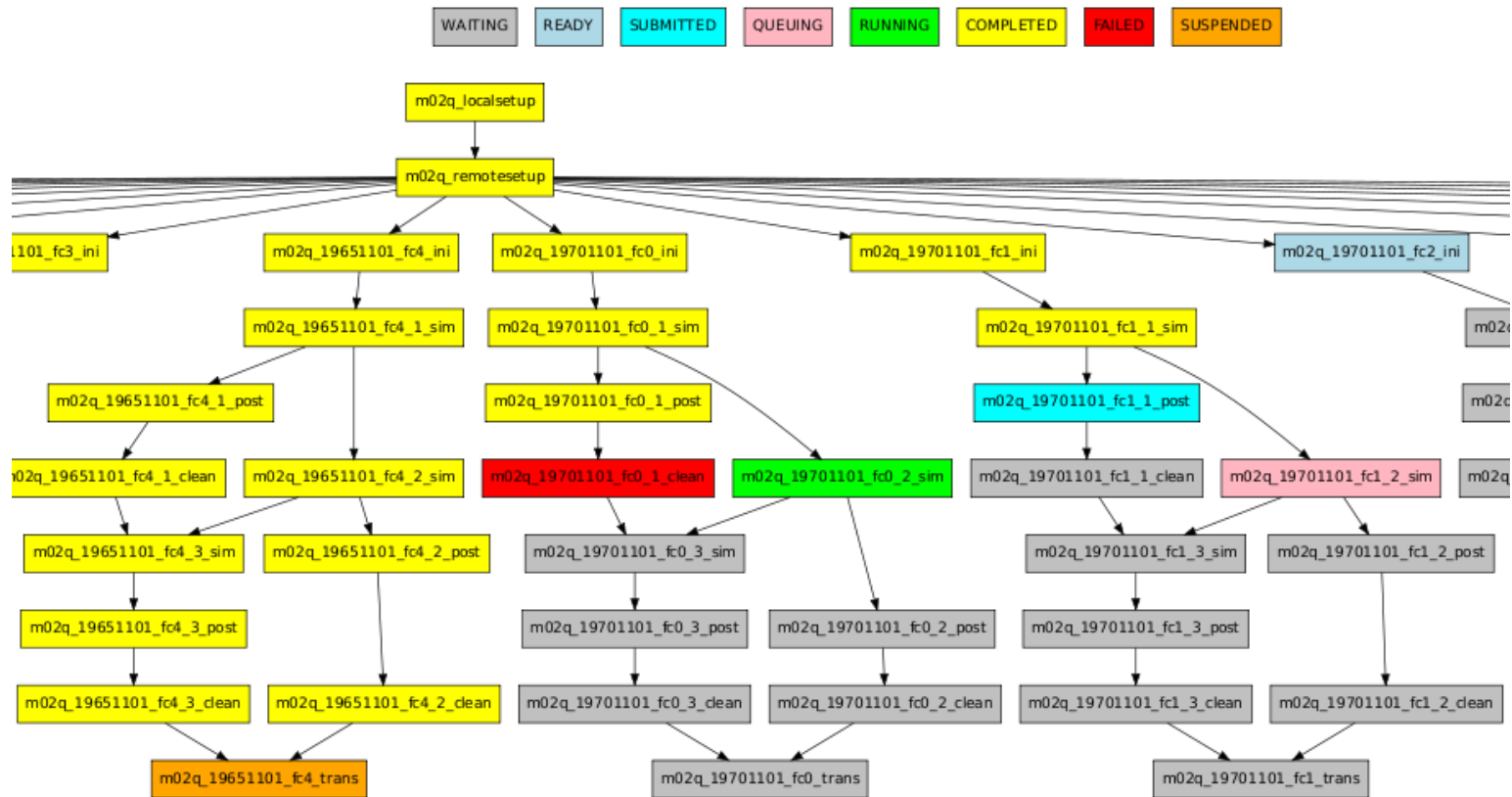


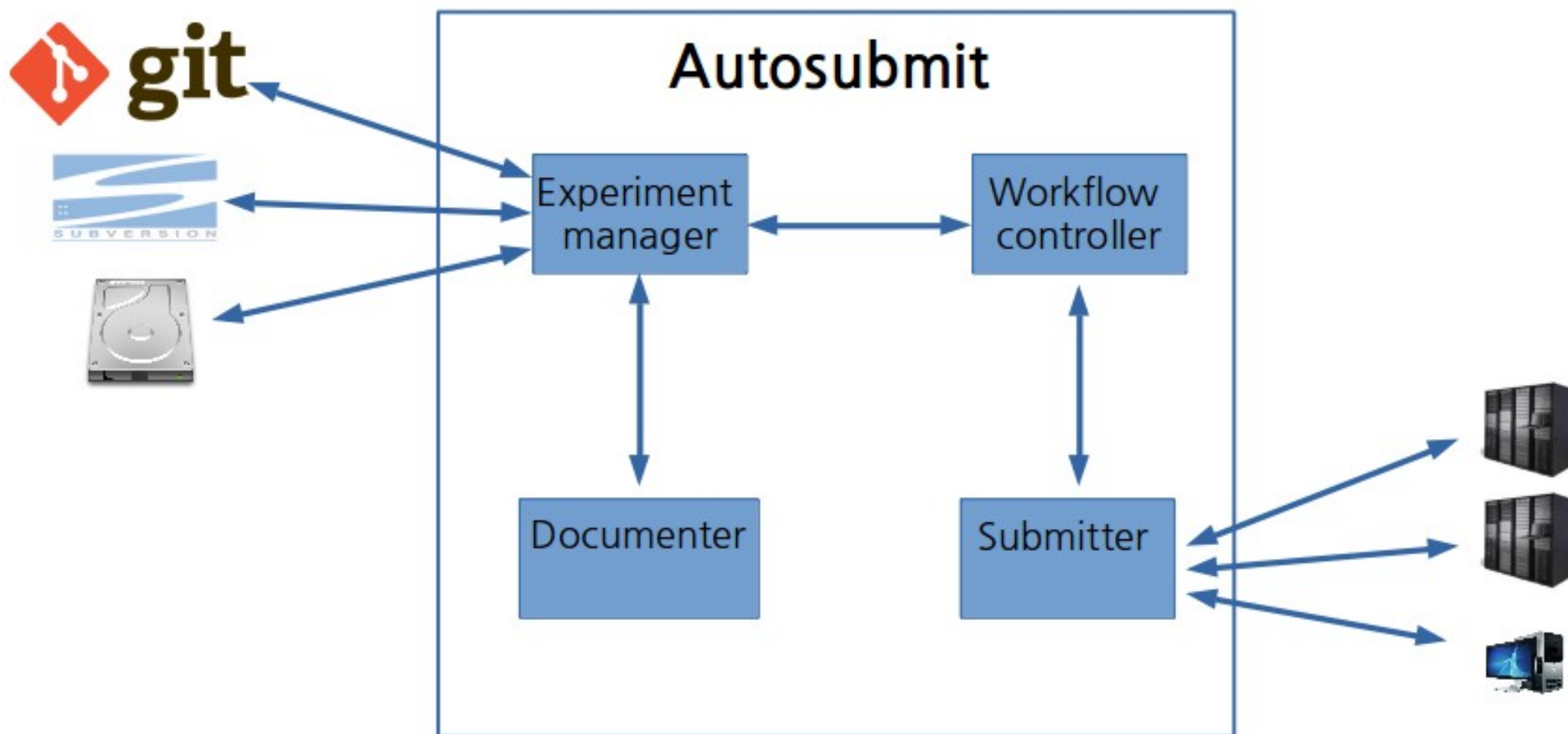
# Autosubmit 3.1



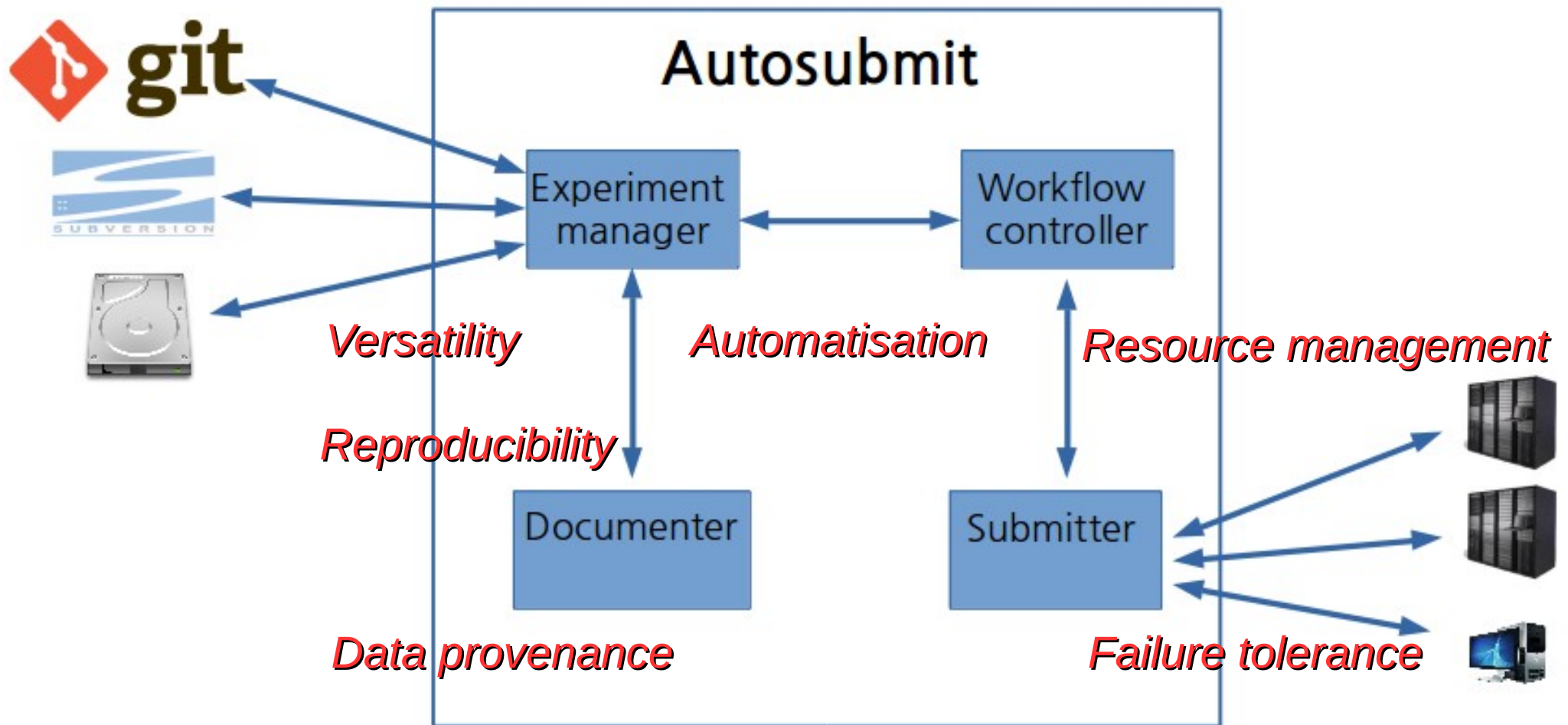
D. Manubens, J. Vegas

O. Mula-Valls, M. Asif, K. Serradell, PA. Bretonnière, V. Guemas, C. Prodhomme,  
O. Bellprat, F. J. Doblas-Reyes

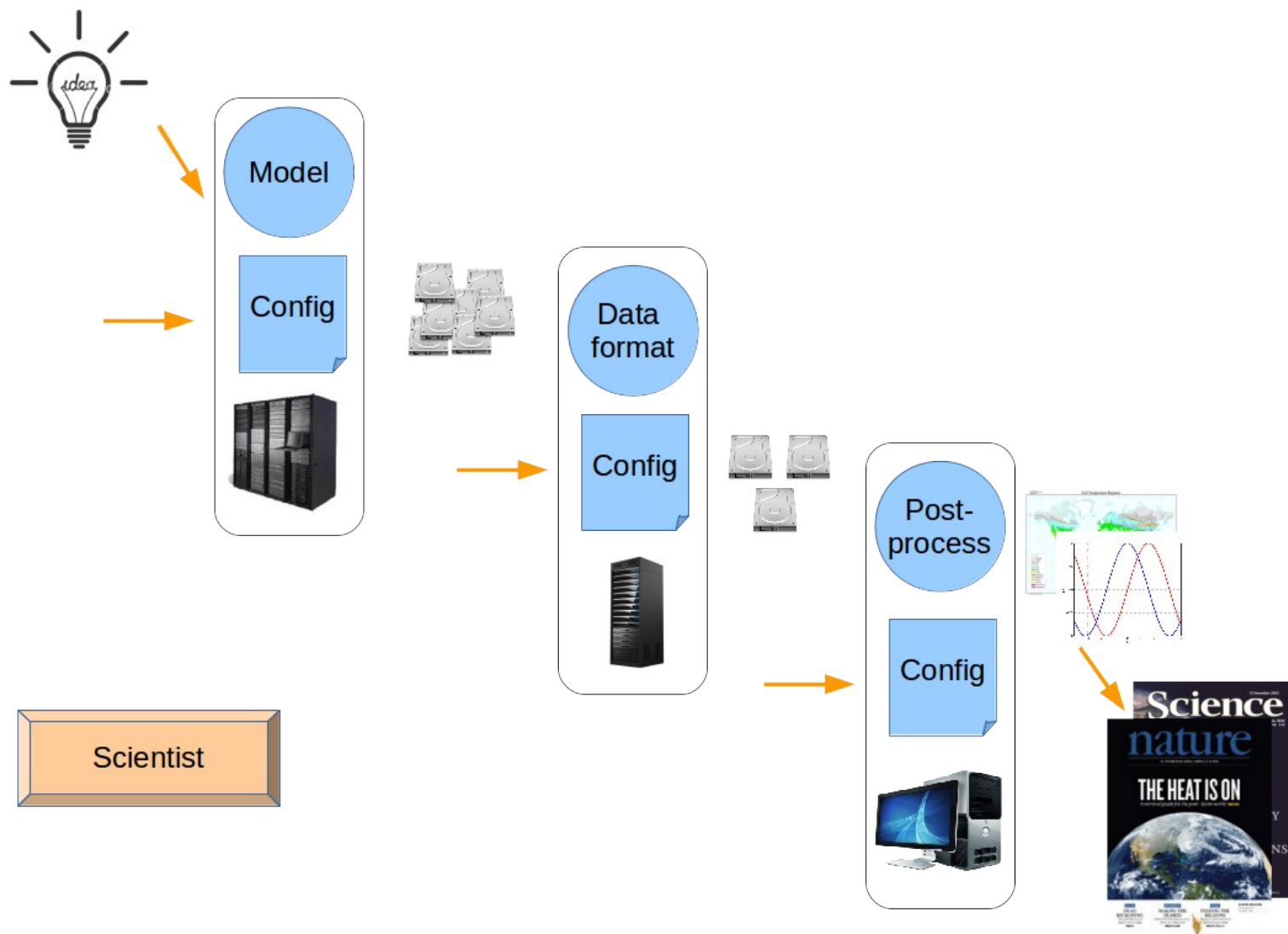
# What is Autosubmit ?



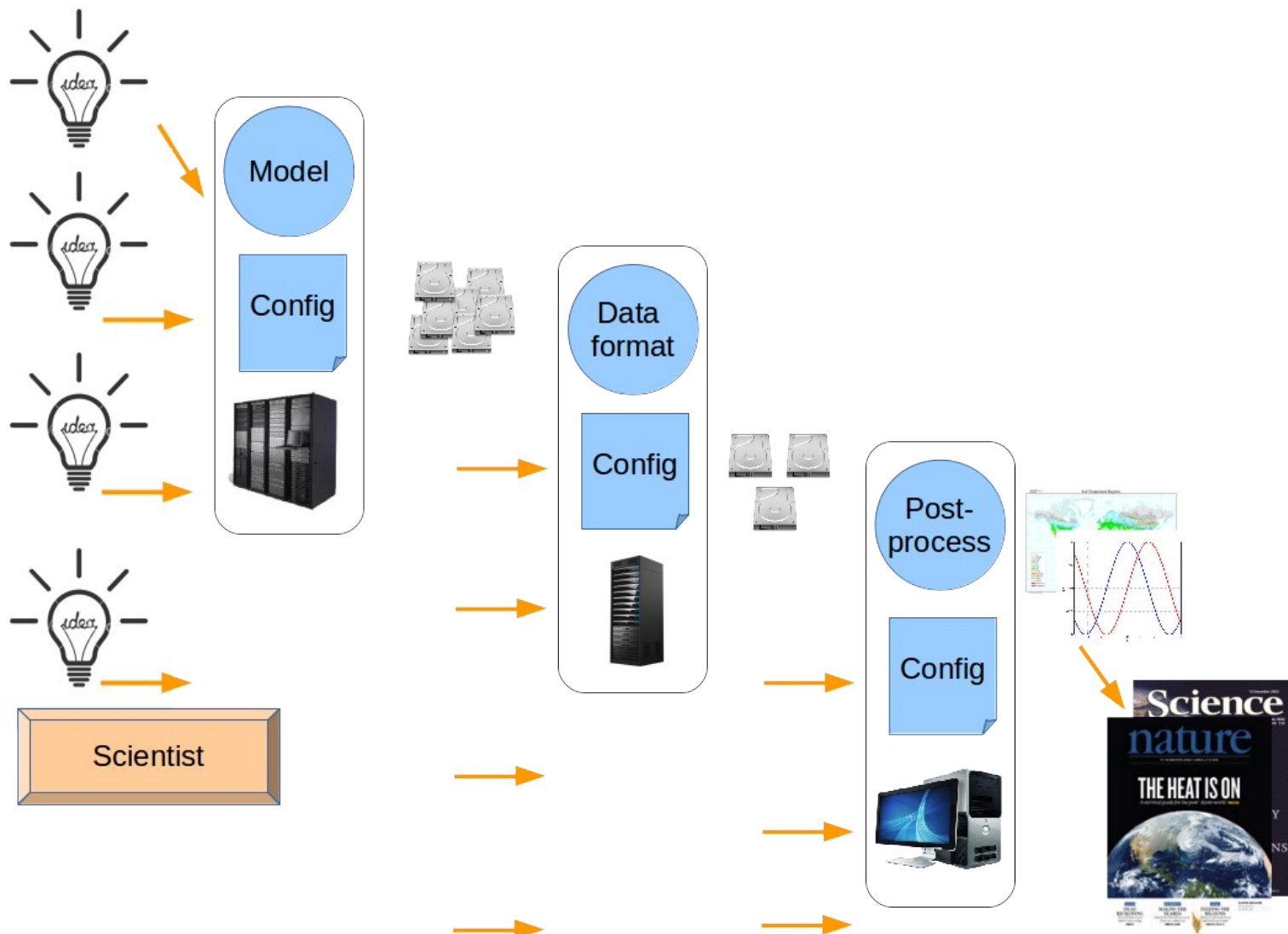
# Why is Autosubmit needed ?



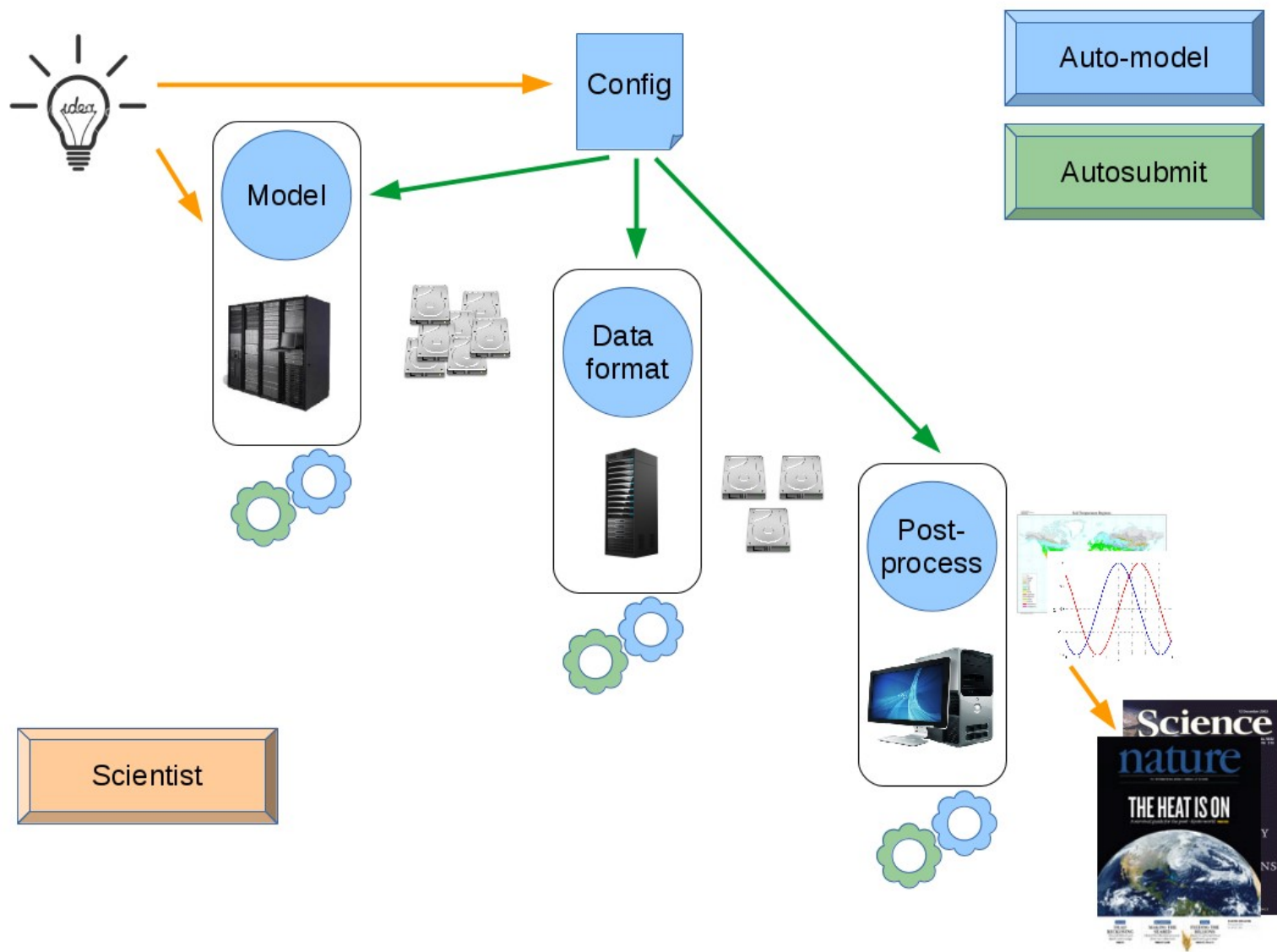
# Why is Autosubmit needed ?



# Why is Autosubmit needed ?



# Why is Autosubmit needed ?



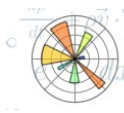
# How does Autosubmit work ?

Python Package Index (PyPI)

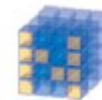
Download  
autosubmit-3.0.0a21.tar.gz



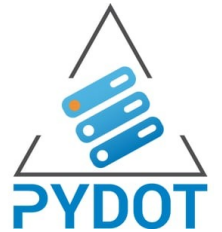
python™ bash.



matplotlib

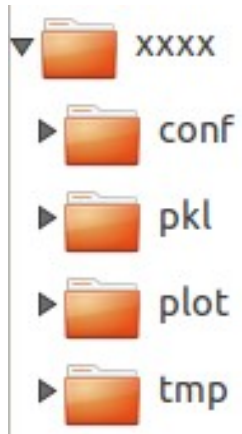


NumPy



## Experiment creation

```
autosubmit expid -H HPCname
```



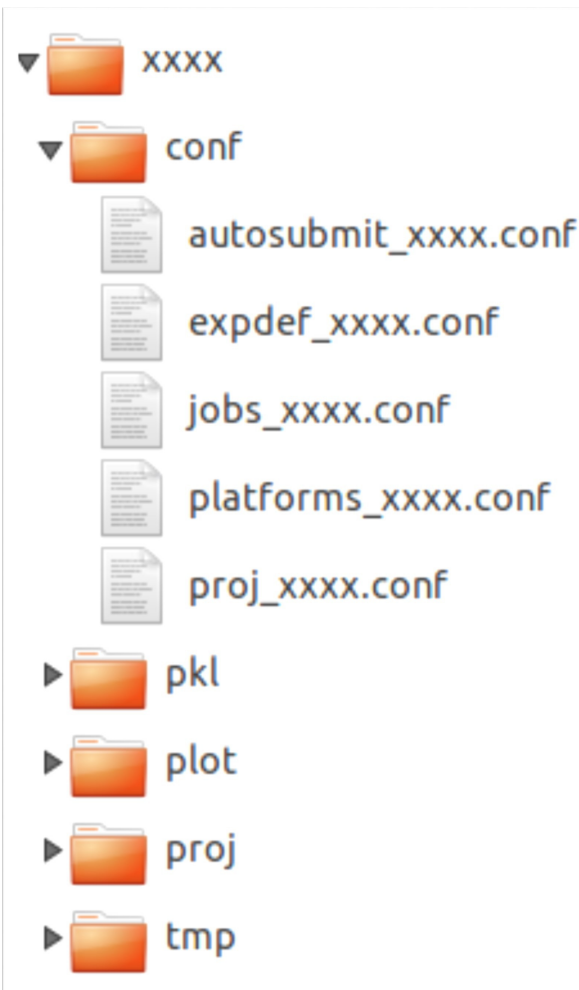


## Experiment creation

## Experiment configuration

```
autosubmit expid -H HPCName
```

```
autosubmit create xxxx
```



Start dates, members and chunks (number and length).

Experiment project source: origin (version control system or path) and project configuration file path.

**expdef\_xxxx.conf**

Workflow to be run: scripts to execute, dependencies between tasks, task requirements (processors, wallclock time...) and platform to use.

**jobs\_xxxx.conf**

HPC, fat-nodes and supporting computers configuration.

Usually provided by technicians, users will only have to change login and accounting options for HPCs.

**platforms\_xxxx.conf**

Project dependant experiment variables that Autosubmit will substitute in the scripts to be run.

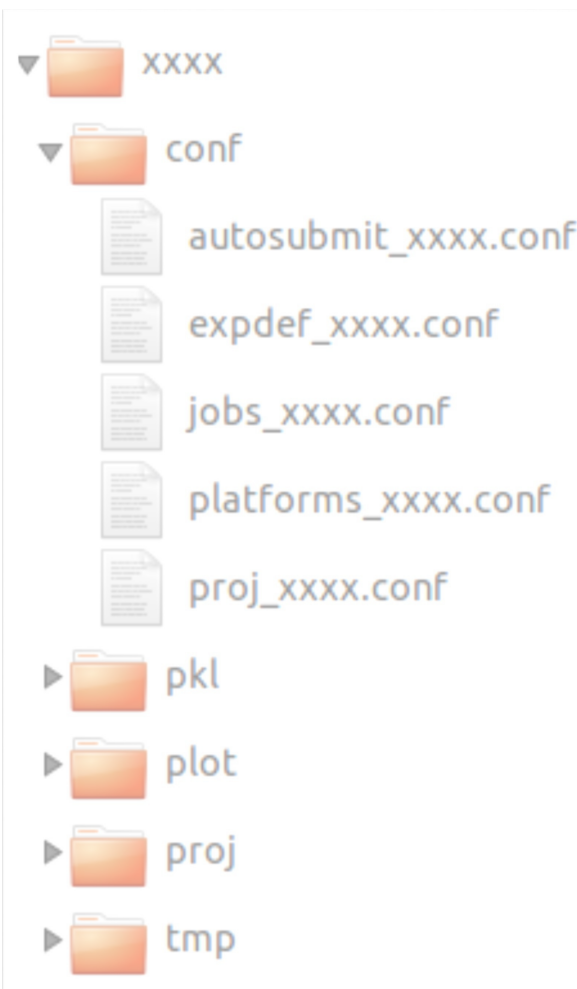
**proj\_xxxx.conf**

## Experiment creation

```
autosubmit expid -H HPCname
```

## Experiment configuration

```
autosubmit create xxxx
```



Start dates, members and chunks (number and length).

Experiment project source: origin (version control system or path) and project configuration file path.

**expdef\_xxxx.conf**

Workflow to be run: scripts to execute, dependencies between tasks, task requirements (processors, wallclock time...) and platform to use.

**jobs\_xxxx.conf**

HPC, fat-nodes and supporting computers configuration.

Usually provided by technicians, users will only have to change login and accounting options for HPCs.

**platforms\_xxxx.conf**

Project dependant experiment variables that Autosubmit will substitute in the scripts to be run.

**proj\_xxxx.conf**

## Experiment configuration file

```
[project]
# Select project type. STRING = git, svn, local, none
# If PROJECT_TYPE is set to none, Autosubmit self-contained dummy templates will be used
PROJECT_TYPE = git
# Destination folder name for project. type = STRING, default = leave empty,
PROJECT_DESTINATION = model

# If PROJECT_TYPE is not git, no need to change
[git]
# Repository URL STRING = 'https://github.com/torvalds/linux.git'
PROJECT_ORIGIN = https://gitlab.cfu.local/cfu/auto-ecearth3.git
# Select branch or tag, STRING, default = 'master',
# help = {'master' (default), 'develop', 'v3.1b', ...}
PROJECT_BRANCH = develop
# type = STRING, default = leave empty, help = if model branch is a TAG leave empty
PROJECT_COMMIT =

# If PROJECT_TYPE is not svn, no need to change
[svn]
# type = STRING, help = 'https://svn.ec-earth.org/ecearth3'
PROJECT_URL =
# Select revision number. NUMERIC = 1778
PROJECT_REVISION =

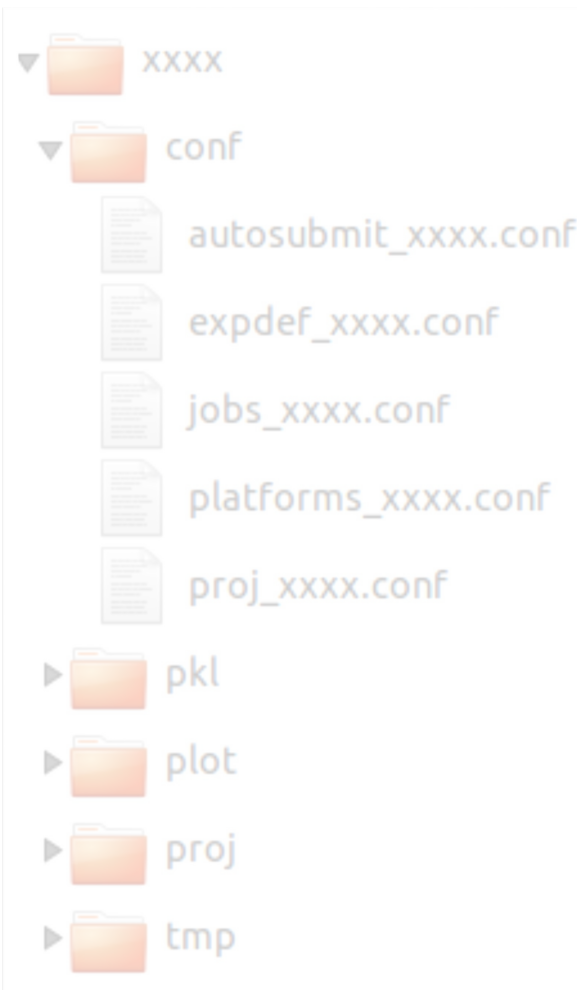
# If PROJECT_TYPE is not local, no need to change
[local]
# type = STRING, help = /foo/bar/ecearth
PROJECT_PATH =

# If PROJECT_TYPE is none, no need to change
[project_files]
# Where is PROJECT CONFIGURATION file location relative to project root path
FILE_PROJECT_CONF = templates/ecearth3/ecearth3.conf
```



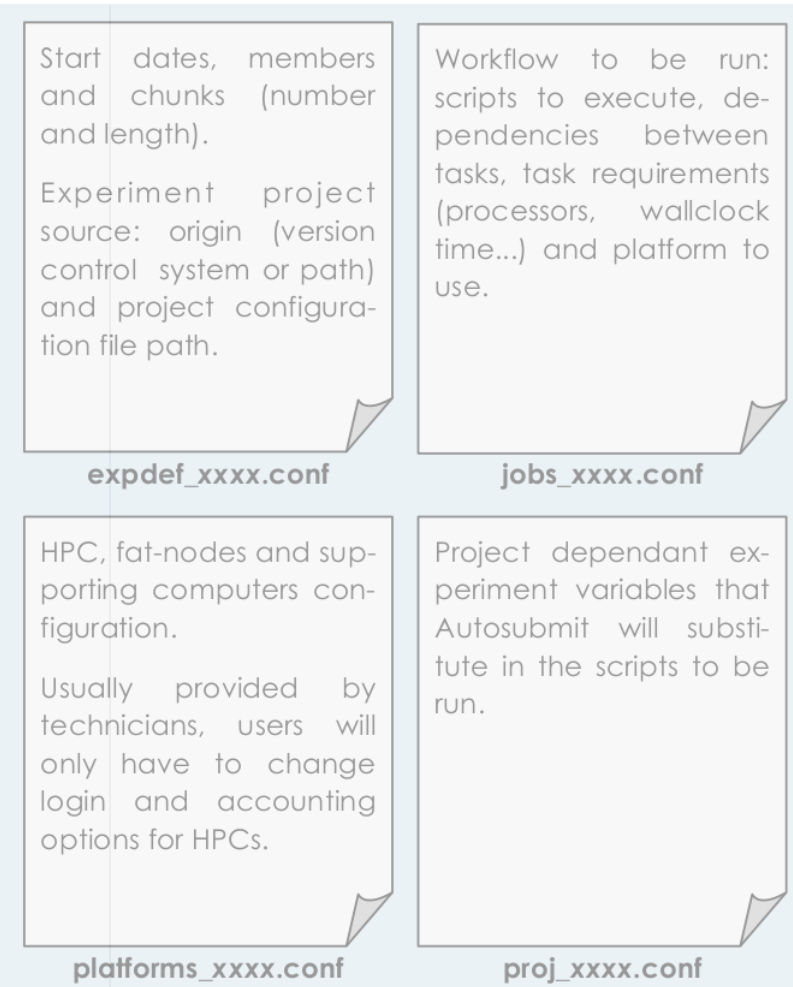
## Experiment creation

```
autosubmit expid -H HPCName
```



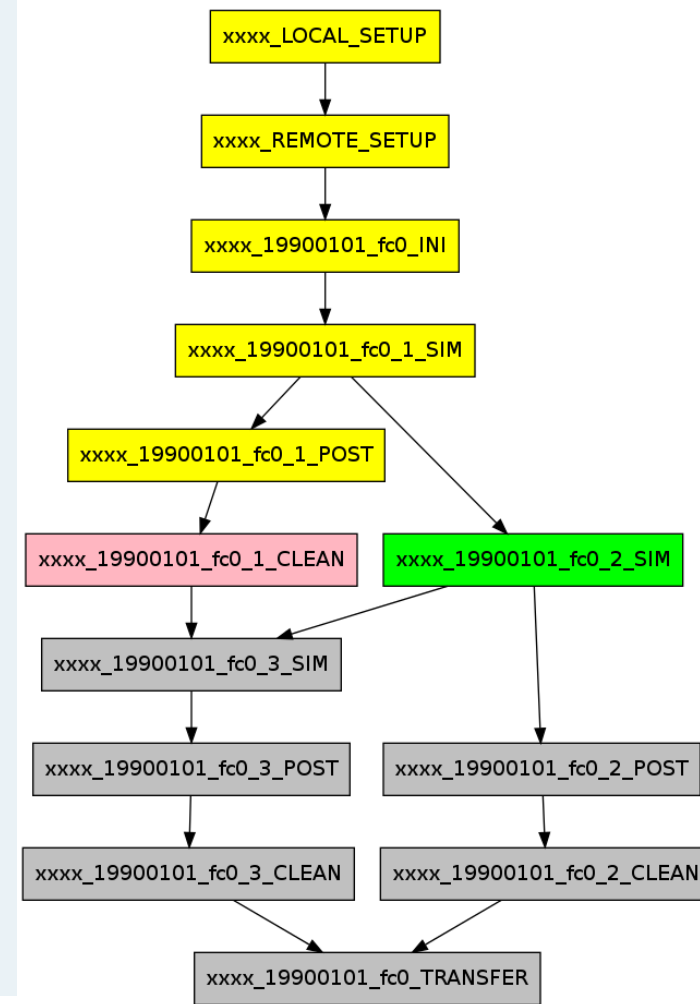
## Experiment configuration

```
autosubmit create xxxx
```



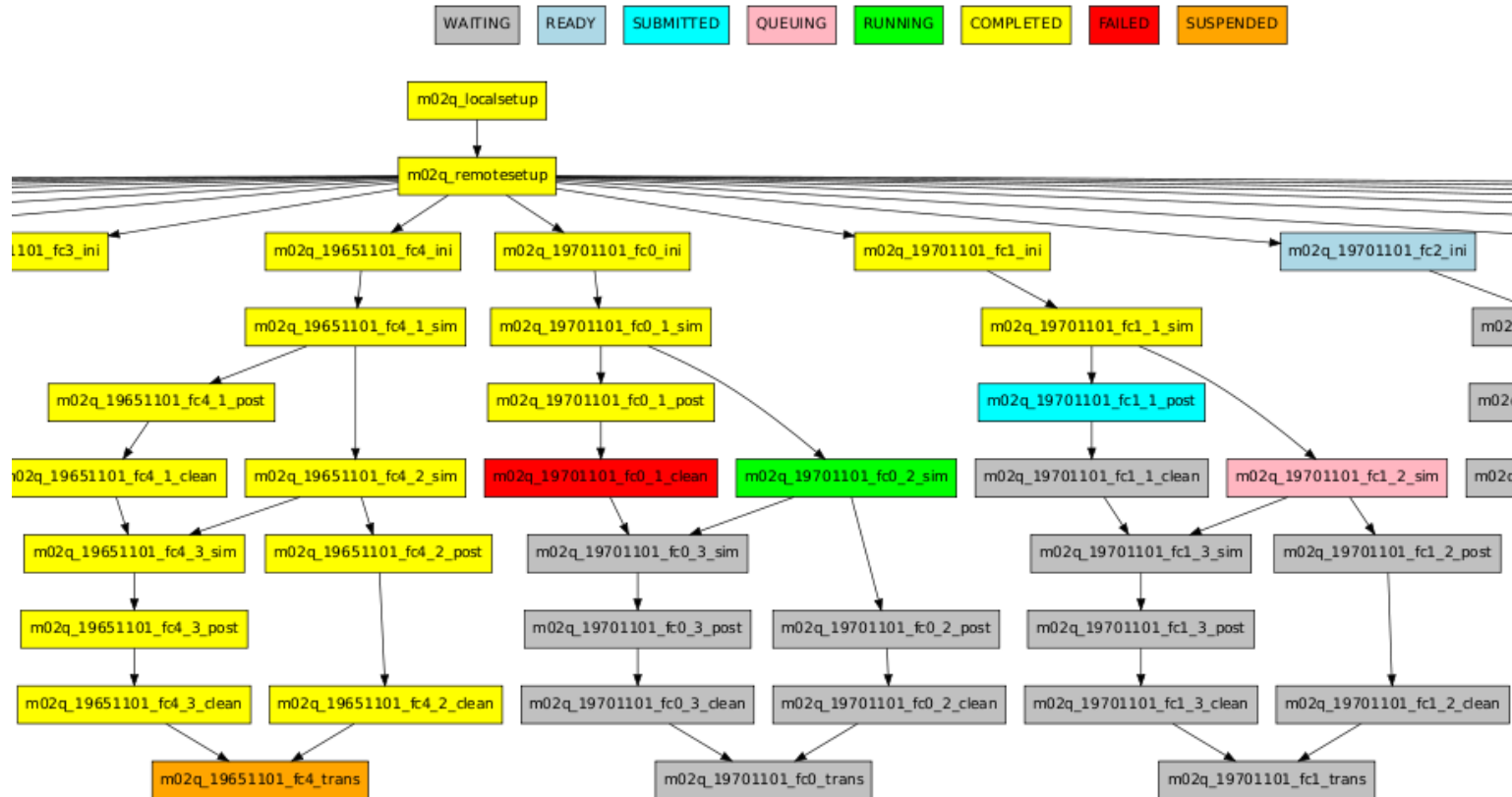
## Experiment run

```
autosubmit run xxxx
```



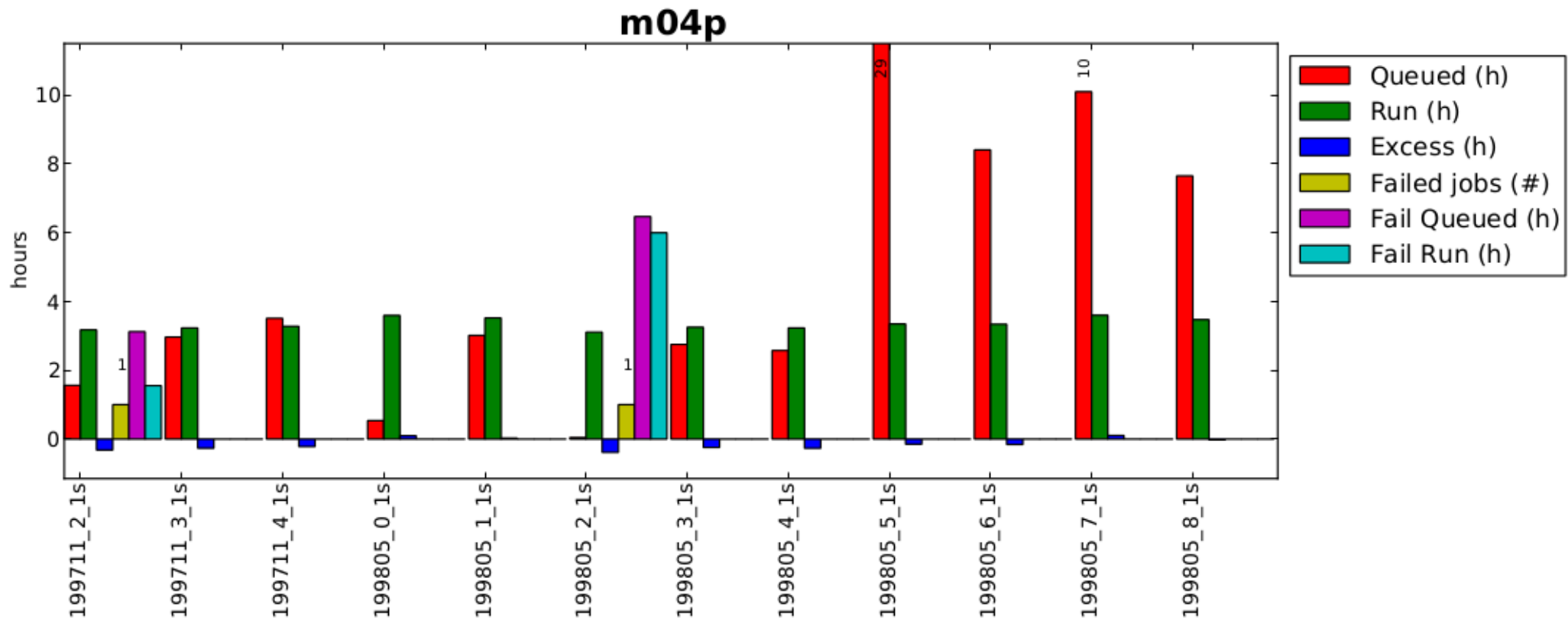
## Experiment monitoring

```
autosubmit monitor xxxx
```

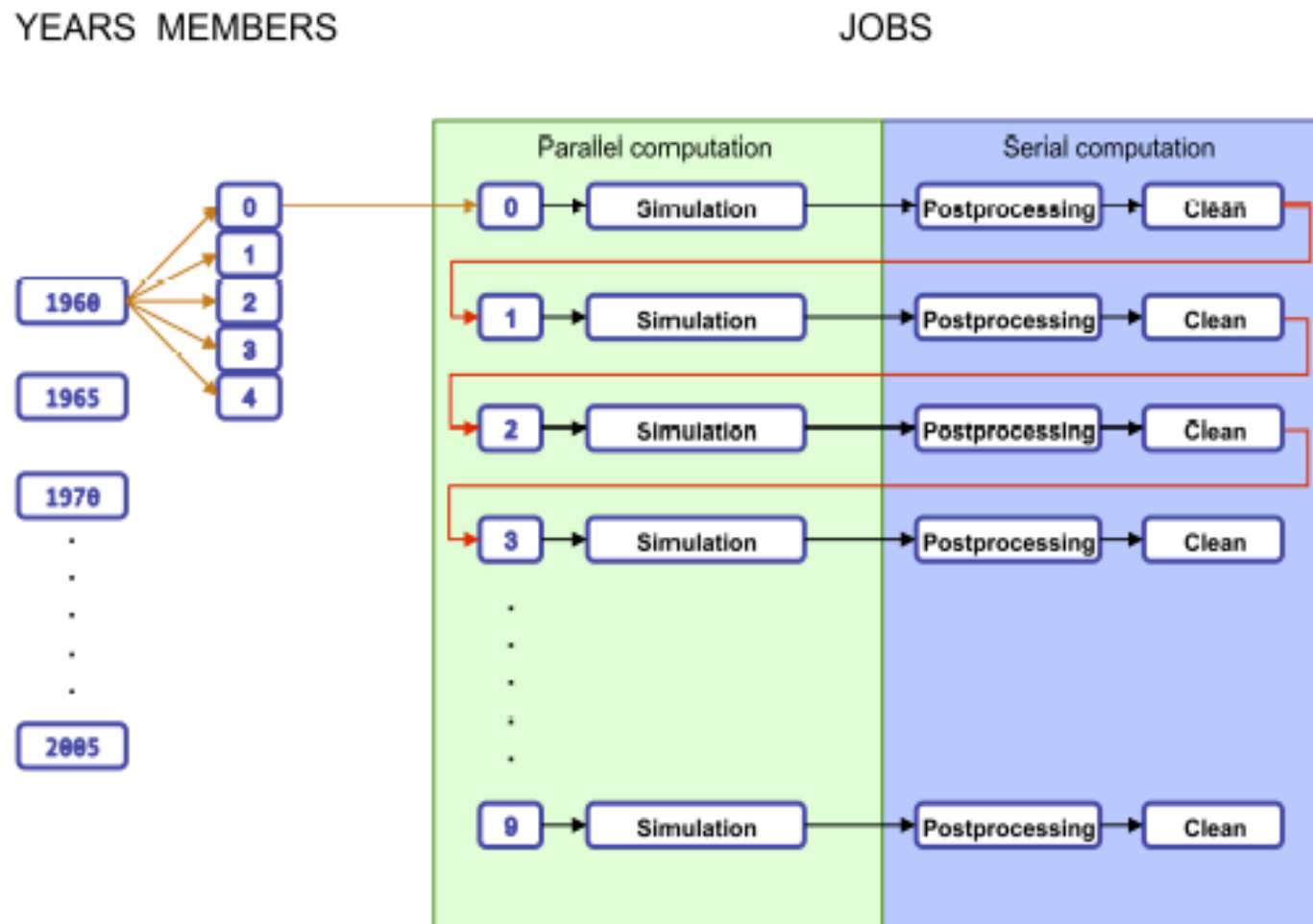


## Automatic statistics

```
autosubmit stats cxxx
```



# EC-Earth/IC3 workflow





# NMMB/BSC workflow

- Currently ~ 1.000 lines of code
- Splited by blocks (fixed, variable, model...)

```
/bin/bash
/bin/bash 113x46
! /bin/bash
set -x

ulimit -s unlimited

#-----
# MAIN NMMB-DUST RUN SCRIPT - Define MN settings first
#-----
INPES=02          # Number inpes
JNPES=03          # Number jnpes
WRTSK=02          # Number write tasks
#-----
# Global-regional switch - Model domain setup global/regional
#-----
DOMAIN=0          # GLOBAL/REGIONAL (0/1)
LM=64             # Vertical model layers
CASE=GL08         # Name of the case
#-----
# If regional you need to modify manually files llgrid_chem.inc in vrbl409rrtm_bsc1.0_reg
#-----
DT_INT1=60        # Run time step (integer seconds) !180
TLMOD1=0.0        # Center point longitudinal (E/W)
TPHOD1=0.0        # Center point latitudinal (S/N)
WBD1=-180.0       # Western boundary (from center point)
SBD1=-90.0        # Southern boundary (from center point)
DLMD1=0.3125      # Longitudinal grid resolution
DPHD1=0.221948    # Latitudinal grid resolution
PTOP1=100.        # Pressure top of the domain (Pa)
DCAL1=0.255       # Mineral Dust Emission Calibration Factor
NRADS1=60         # Number of timesteps between radiation calls (short)
NRADL1=60         # Number of timesteps between radiation calls (long)
#-----
DT_INT2=30        # regional
TLMOD2=20.0       # regional
TPHOD2=35.0       # regional
WBD2=-51.0        # regional
SBD2=-35.0        # regional
DLMD2=0.30        # regional
DPHD2=0.30        # regional
PTOP2=5000.       # Pressure top of the domain (Pa)
DCAL2=0.255       # Mineral Dust Emission Calibration Factor
NRADS2=120        # Number of timesteps between radiation calls (short)
NRADL2=120        # Number of timesteps between radiation calls (long)
#-----
# Case selection
```

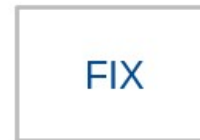


# Splitting tasks

- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST
- And we define dependences

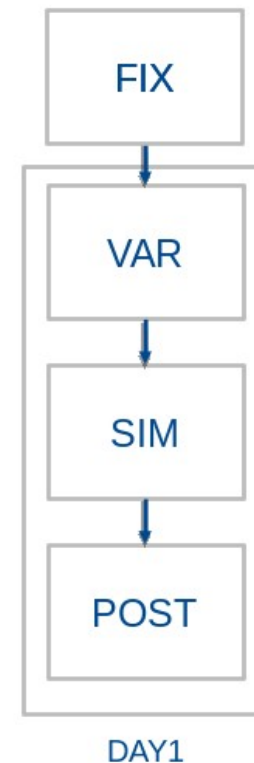
# Splitting tasks

- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST
- And we define dependences



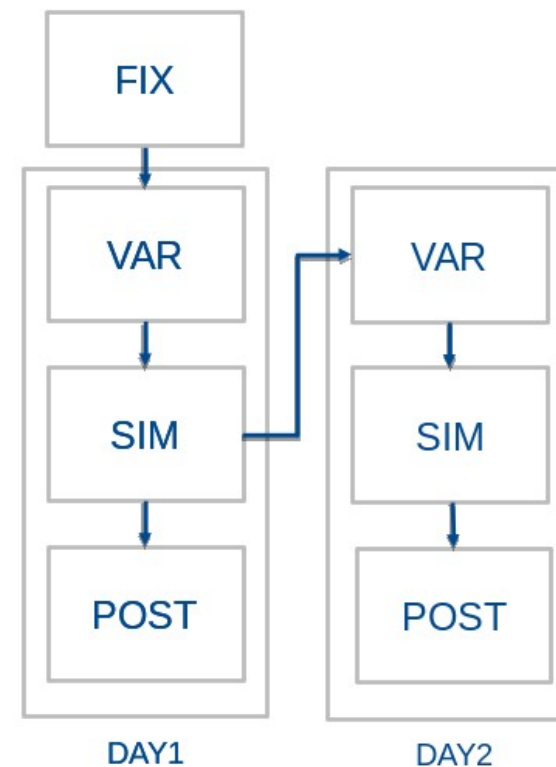
# Splitting tasks

- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST
- And we define dependences



# Splitting tasks

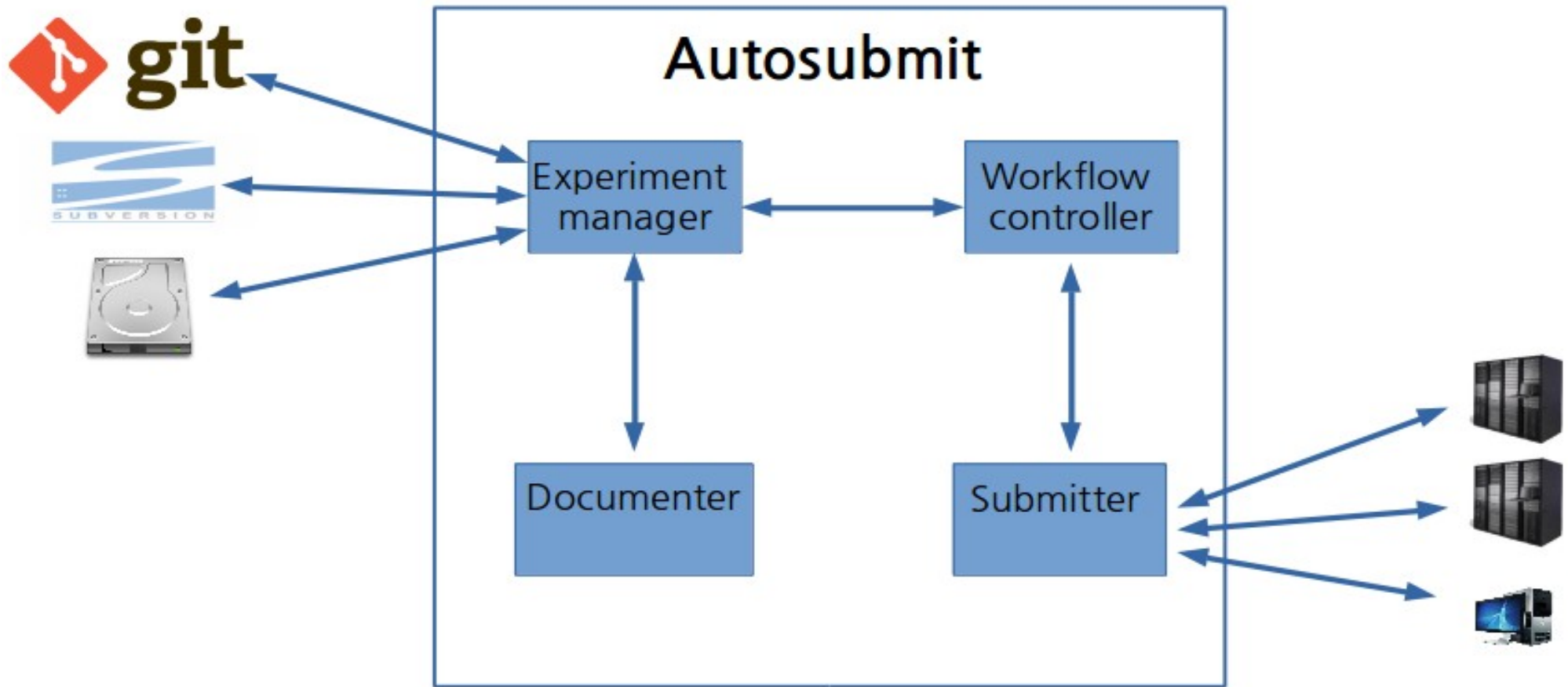
- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST
- And we define dependences



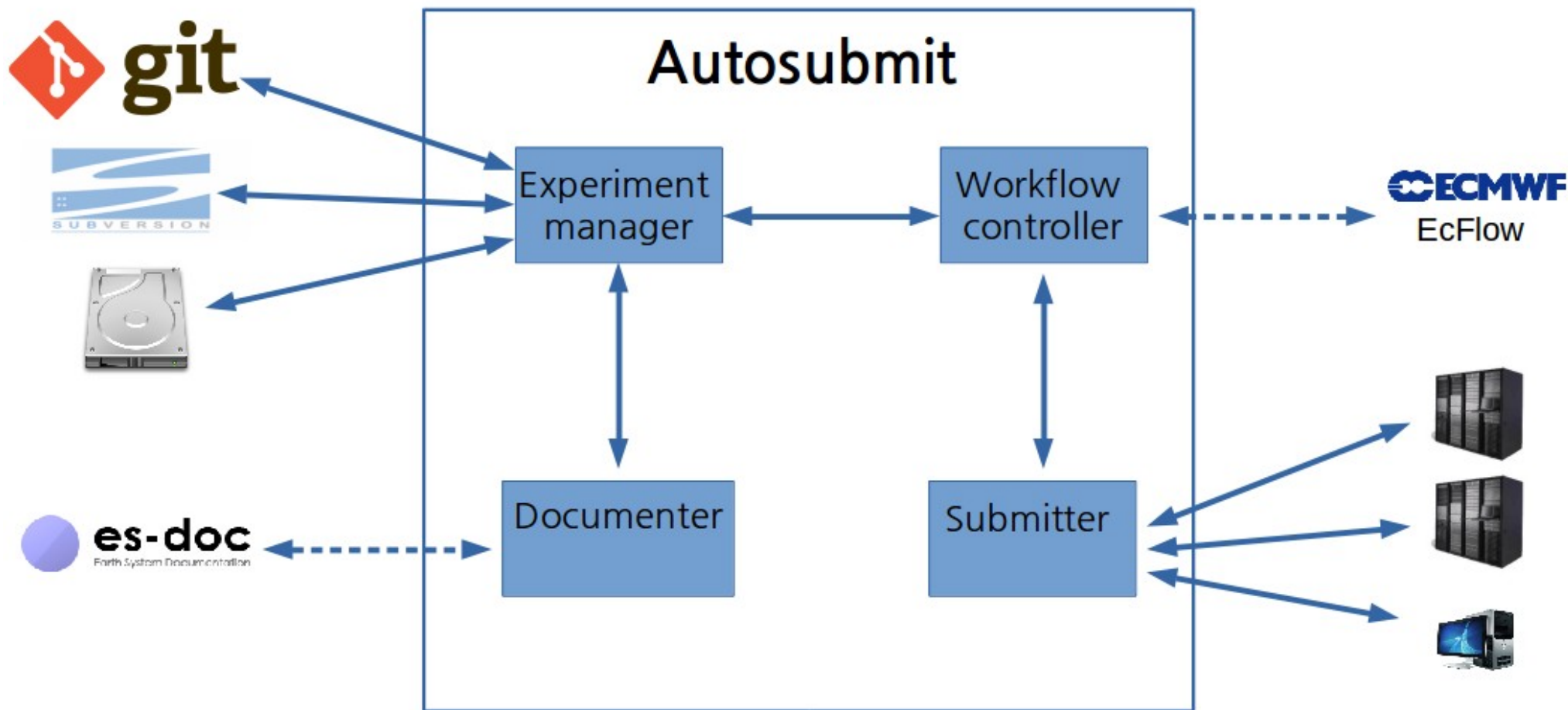
# Future Work

- Create a wrapper for jobs that allows to send various independent tasks as one big job to reduce queueing time and to allow participation in tier-0 projects.
- Implement the SAGA standard for job submission & control.
- Expand data stored in database to store all data needed for CMIP6. Use the es-doc python API.
- Build a plugin system for different models databases.
- Evaluate the suitability of using ecFlow (<https://software.ecmwf.int/wiki/display/ECFLOW>) and Cylc (<http://cylc.github.io/cylc>) for workflow configuration to converge with community standards.

# Future Work



# Future Work



# Roadmap

